

Simulasi Numerik untuk Menentukan Gaya Gravitasi Bawah Permukaan Pada Model Prisma Segiempat

Indra Gunawan¹, Susanti Alawiyah²

Teknik Geofisika, Institut Teknologi Bandung
Jl. Ganesha 10, Gedung BSC-B, Lantai 2, Bandung, Indonesia

¹ gunawan@geoph.itb.ac.id

² santi@gf.itb.ac.id

Abstract

Resolution of the subsurface model using gravity data with the rectangular prisms approach are determined by the size of the prisms. The smaller size of the prisms, the higher resolution will be obtained. Unfortunately, the size of the prisms are directly correlated with the amount of prisms at a model volume. Number of prisms also be associated with the computing price for the area to calculate. To raise this matter, we have applied forward modeling calculation for a rectangular prism with homogeneous density for several methods. We analyzed the efficiency, accuracy and capability of each method for double precision calculations with C++ language. As the sample object and observation points, we follows the standard cubic model to comparing our results. The results show that Okabe methods has the best efficiency, it is 290.70% compare with the longest time. Okabe method also has good accuracy with the correlation factor near 1 when it compared by reference. This method, with Haaz-Jung-Plouff and Sorokin also could elude some error calculation which are caused by singularity properties. We confirm that some methods couldn't calculate the model at some observation points due to singularity. In this paper, we give the benchmark information and review of the methods based on our tests.

Keywords: Benchmark, forward modeling method, gravity calculation, subsurface density.

Abstrak

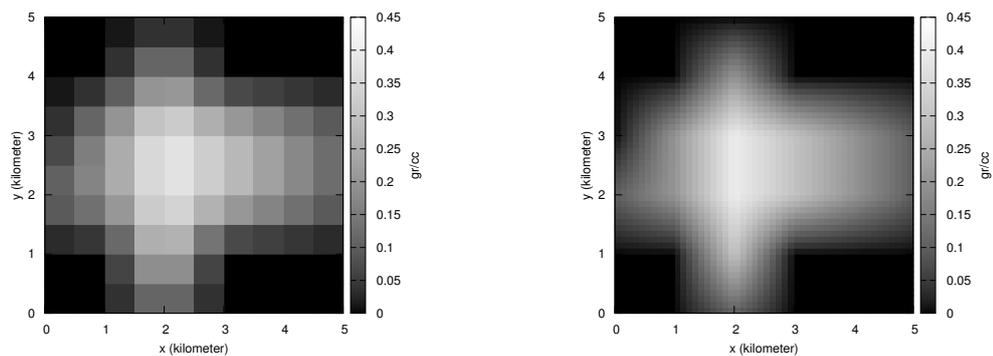
Resolusi model bawah permukaan menggunakan data gaya berat dengan pendekatan prisma segiempat ditentukan oleh ukuran dari prisma tersebut. Semakin kecil ukuran prisma, maka semakin tinggi resolusi yang dihasilkan. Sayangnya, ukuran dari prisma berkorelasi langsung dengan jumlah prisma pada volume model. Jumlah dari prisma juga akan berhubungan dengan harga komputasi yang harus dibayar untuk menghitung area yang sedang dikaji. Untuk mengatasi hal ini, dalam penelitian ini digunakan beberapa metode perhitungan *forward modeling* melalui pendekatan prisma segiempat sebagai akibat dari model objek massa jenis yang homogen. Selanjutnya dilakukan analisa perbandingan terhadap efisiensi, akurasi dan kemampuan dari setiap metode menggunakan bahasa pemrograman C++ dengan presisi ganda. Sebagai contoh objek dan titik observasi, digunakan *standard cubic model* untuk membandingkannya dengan hasil perhitungan. Hasil yang diperoleh didapatkan bahwa metode Okabe memiliki tingkat efisiensi waktu paling baik, yaitu sebesar 290.70% jika dibandingkan dengan metode perhitungan terlama. Metode Okabe juga memiliki akurasi yang sangat baik dengan tingkat korelasi mendekati 1 ketika dibandingkan dengan referensi. Metode ini, bersama metode Haaz-Jung-Plouff dan Sorokin juga mampu menghindari perhitungan error paling banyak yang diakibatkan oleh sifat *singularity*. Kami mengkonfirmasi bahwa ada beberapa titik observasi yang tidak bisa dihitung karena adanya sifat *singularity*. Paper ini memberikan informasi *benchmark* dan *review* metode berdasarkan pada pengujian yang dilakukan.

Kata Kunci: Benchmark, massa jenis bawah permukaan, metode *forward modeling*, perhitungan nilai gravitasi.

I. PENDAHULUAN

SALAH satu metode geofisika untuk mendekati model massa jenis bawah permukaan adalah metode gaya berat. Metode gaya berat menggunakan respons gaya berat yang diukur oleh gravimeter untuk diolah menjadi model massa jenis bawah permukaan, karena pada prinsipnya setiap benda yang memiliki massa jenis akan memberikan pengaruh gaya berat pada daerah disekitarnya. Pengolahan data gaya berat menjadi model massa jenis membutuhkan teknik pemodelan inversi yang berkaitan erat dengan metode *forward modeling* [1]–[4]. *Forward modeling* dilakukan secara iteratif dalam setiap langkah inversi, ini membuat harga komputasi pada teknik inversi menjadi mahal. Permasalahan harga komputasi bisa diakomodasi dengan meningkatkan kualitas perangkat keras, namun untuk mendapatkan efisiensi yang maksimal diperlukan juga teknik komputasi yang efisien.

Efisiensi teknik komputasi pada *forward modeling* dipengaruhi oleh beberapa faktor, diantaranya adalah model geometri massa jenis, model persamaan matematika, teknik paralelisasi, serta bahasa pemrograman yang digunakan. Ada beberapa cara untuk mendekati model geometri massa jenis bawah permukaan. Bentuk yang umum dipakai adalah menggunakan pendekatan titik-massa, prisma segiempat, prisma segibanyak, silinder, serta metode elemen hingga. Bentuk tersebut memiliki kesamaan, yaitu memiliki harga komputasi yang berbanding lurus dengan tingginya resolusi hasil yang ingin diperoleh. Resolusi peta massa jenis bawah permukaan ditentukan oleh besar kecilnya model geometri pada sebuah volume di sistem yang sedang dihitung (Gambar 1).



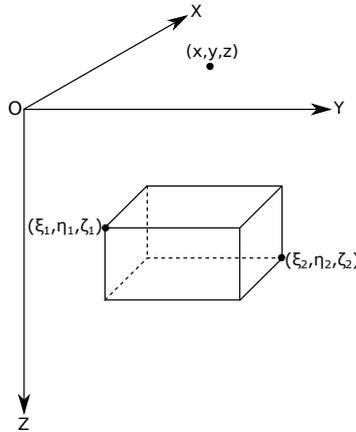
(a) Peta massa jenis resolusi rendah.

(b) Peta massa jenis resolusi tinggi.

Gambar 1: Peta massa jenis yang ditunjukkan oleh blok prisma segiempat (dilihat vertikal dari atas). Gambar di sebelah kiri (a) merupakan peta massa jenis dengan resolusi rendah yang memiliki ukuran prisma lebih besar dan sedikit, sedangkan gambar di sebelah kanan (b) merupakan peta massa jenis dengan resolusi tinggi yang memiliki ukuran prisma lebih kecil dan banyak.

Pada penelitian ini, dilakukan kajian terhadap bentuk geometri prisma segiempat (Gambar 2) dengan sebaran nilai massa jenis homogen di dalam prisma. Koordinat bodi disimbolkan oleh (ξ, η, ζ) , sedangkan koordinat stasiun disimbolkan oleh (x, y, z) . Dengan menggunakan bentuk geometri tersebut kami membandingkan lima model matematika yang umum digunakan pada *forward modeling*. Kelima metode tersebut diantaranya adalah metode Sorokin op. cit. [5], Haaz-Jung-Plouff [6] op. cit. [5], Okabe op. cit. [5], Nagy op. cit. [5] dan Zhou [7]. Semua metode tersebut dikaji dengan membandingkan waktu komputasi, tingkat akurasi dan kemampuan tiap-tiap metode untuk mendapatkan hasil pada berbagai keadaan posisi stasiun percobaan. Sebagai contoh objek dan posisi stasiun observasi, digunakan *standard cubic model* dan beberapa posisi stasiun tambahan yang digunakan oleh Li dan Chouteau [5]. Model ini juga telah diadopsi oleh Commer [8] untuk memperlihatkan sebaran nilai percepatan gaya berat secara umum. Penelitian sebelumnya oleh Li dan Chouteau [5] serta Commer [8] telah membandingkan kestabilan perhitungan keempat metode *forward modeling* (yaitu: Sorokin, Haaz-Jung-Plouff, Okabe dan Nagy) dan sebaran posisi yang menghasilkan error numerik sebagai akibat dari adanya *singularity*. Namun sayangnya, mereka belum menjabarkan efisiensi dari masing-masing metode perhitungan.

Paper ini memberikan informasi mengenai benchmark dan ulasan metode *forward modeling* berda-



Gambar 2: Model prisma segiempat.

sarkan pengujian yang dilakukan. Benchmark dilakukan dengan melihat perbandingan kecepatan masing-masing metode untuk menyelesaikan perhitungan, membandingkan akurasi dengan hasil referensi dan melihat kemampuan masing-masing metode untuk menghitung segala kemungkinan posisi titik stasiun observasi. Setelah itu ulasan akan diberikan berdasarkan hasil yang diperoleh dan menjelaskan hal yang menyebabkan hasil tersebut. Penelitian ini belum memperhitungkan efek dari parallelisasi dan melakukan perbandingan bahasa pemrograman yang digunakan. Dalam kajian ini baru dilakukan perhitungan pada mesin tunggal tanpa parallelisasi dan menggunakan bahasa pemrograman C++ dengan presisi ganda serta tambahan *library* GSL [9] dan Boost [10] untuk membantu perhitungan integrasi numerik.

II. DASAR TEORI

Forward modeling merupakan istilah untuk menghitung besarnya percepatan gravitasi pada titik tertentu sebagai akibat dari keberadaan sebuah benda. *Forward modeling* berlandaskan pada hukum gravitasi umum Newton dimana besarnya gaya tarik-menarik antara dua benda (\vec{F} dalam satuan *Newton*) berbanding lurus dengan besarnya massa kedua benda tersebut (m dan M dalam satuan *kilogram*) dan berbanding terbalik dengan kuadrat dari jarak keduanya (r^2 dalam satuan *meter*).

$$\vec{F} = -G \frac{mM}{r^2} \hat{r} \tag{1}$$

dimana G adalah konstanta gravitasi $6.674 \times 10^{-11} \text{ Nm}^2/\text{kg}^2$.

Dari persamaan (1), besarnya percepatan gravitasi (\vec{g} dalam satuan m/s^2) pada sebuah titik acuan karena keberadaan benda bermassa (m) pada jarak tertentu (\vec{r}) secara sederhana dapat dituliskan menjadi persamaan (2).

$$\vec{g}(r) = -G \frac{m}{r^2} \hat{r} \tag{2}$$

Karena massa benda (m) bergantung pada massa jenisnya (ρ dalam satuan kg/m^3) yang didefinisikan sebagai perkalian antara massa jenis benda dengan volume (V dalam satuan m^3) nya, maka persamaan (2) dapat dituliskan juga menjadi persamaan (3).

$$\vec{g}(r) = -G \frac{\rho V}{r^2} \hat{r} \tag{3}$$

Persamaan (3) adalah pendekatan *forward modeling* umum yang bisa langsung digunakan pada pendekatan objek geometri berupa bola pejal. Objek geometri berupa bola pejal sulit mengakomodasi kebutuhan perhitungan geometri benda yang lebih rumit, misalnya bentuk geometri berupa prisma segiempat. Pendekatan *forward modeling* menggunakan objek berupa prisma segiempat telah dikembangkan sejak tahun 1800-an, namun solusi persamaan yang paling sering dipakai adalah yang diturunkan oleh Sorokin (1951), Haaz (1953), Jung (1961), Plouff (1966, 1976), Nagy (1966), Okabe (1979), Steiner dan Zilahi-Sebes (1988) serta Zhou (2009). Secara umum, persamaan *forward modeling* untuk objek geometri prisma segiempat (Gambar 2) adalah seperti yang ditunjukkan pada persamaan (4).

$$\vec{g}(x, y, z) = -G\rho \int_{\zeta_1}^{\zeta_2} \int_{\eta_1}^{\eta_2} \int_{\xi_1}^{\xi_2} \frac{z - \zeta}{r^3} d\xi d\eta d\zeta \quad (4)$$

III. DESKRIPSI PENGUJIAN

Dalam penelitian dilakukan pengujian beberapa metode *forward modeling* yang mengikuti pendekatan geometri prisma segiempat dengan nilai sebaran massa jenis yang homogen. Metode yang dikaji yaitu yang diturunkan oleh Sorokin, Haaz-Jung-Plouff, Okabe, Nagy dan Zhou menggunakan persamaan (4).

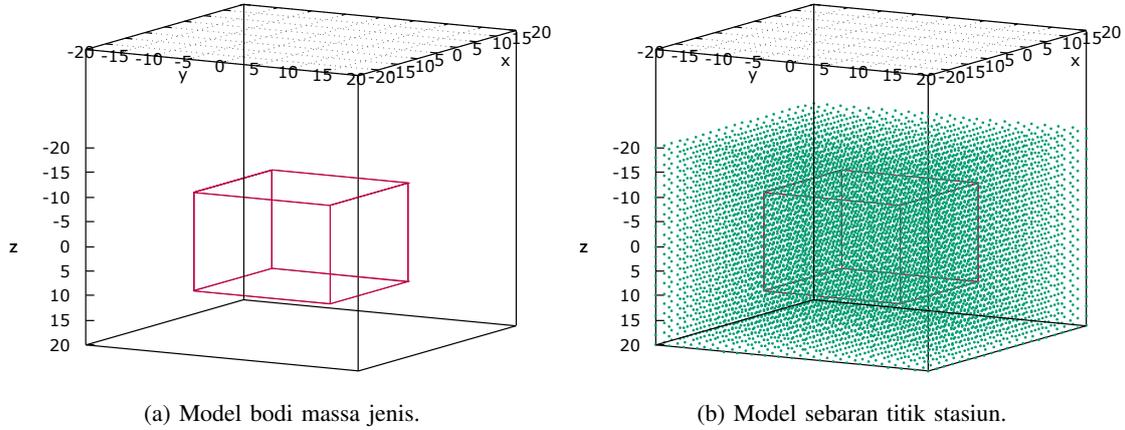
Metode pengujian dibagi menjadi tiga aspek yaitu:

- 1) Efisiensi. Pada bagian ini pengujian dilakukan untuk membandingkan kecepatan dari masing-masing metode dalam menyelesaikan perhitungan pada semua titik stasiun observasi (9261 titik). Waktu komputasi adalah waktu yang dibutuhkan untuk menghitung percepatan gravitasi sebagai akibat dari satu bodi massa jenis pada satu titik acuan. Waktu yang dibutuhkan untuk membaca file input dan pengisian matrix tidak diperhitungkan.

Kode Program:

```
for (i=0; i<nStations; i++)
{
    gettimeofday(&start, NULL);
    cout << forward_model(bodies, nBodies, stations[i]) << endl;
    gettimeofday(&end, NULL);
    long long time = (end.tv_sec * (unsigned int)le6
                    + end.tv_usec) - (start.tv_sec
                    * (unsigned int)le6 + start.tv_usec);
}
```

- 2) Akurasi. Hasil perhitungan yang diperoleh pada penelitian ini dibandingkan dengan hasil perhitungan yang diperoleh dari penelitian sebelumnya [5]. Ada tambahan 18 titik stasiun observasi untuk memberikan kemungkinan stasiun observasi berada pada posisi paling ekstrem dari 9261 titik yang telah ada. Tambahan titik stasiun observasi ini digunakan untuk melihat efek dari presisi angka *floating point* yang mampu dihitung oleh *compiler*. Penelitian sebelumnya melakukan perhitungan menggunakan bahasa pemrograman Fortran dengan presisi tunggal dan ganda, sedangkan penelitian yang kami lakukan menggunakan bahasa pemrograman C++ dan hanya menggunakan presisi ganda.
- 3) Kapabilitas. Setiap metode diuji untuk dilihat kemampuannya dalam menghitung beberapa kemungkinan posisi stasiun. Berdasarkan persamaan yang dipakai, dapat dilihat bahwa ada beberapa titik stasiun observasi yang tidak bisa dihitung oleh metode tertentu karena adanya sifat *singularity* sehingga terjadi error numerik, yaitu ketika terjadi pembagian bilangan dengan bilangan nol ($f(x) = \frac{1}{x}$ dimana $x = 0$). Sifat *singularity* terjadi pada saat argumen *ln* dan *arctan* pada persamaan (5), (6), (7), (8), dan (9) menghasilkan nilai infinity. Ada beberapa kemungkinan yang menyebabkan hal tersebut bisa terjadi. Untuk penjelasan lebih lanjut mengenai terjadinya *singularity* dapat merujuk pada artikel yang ditulis oleh Commer [8].



Gambar 3: Gambar bodi massa jenis dan sebaran stasiun pada sistem. (a) Model bodi massa jenis digambarkan oleh prisma berwarna merah dengan dimensi 20 x 20 x 20 meter dan memiliki massa jenis 1 gr/cc. (b) Model stasiun digambarkan oleh titik berwarna hijau dengan jumlah 9261 titik yang tersebar pada jarak 2 meter satu sama lain terhadap sumbu x, y dan z.

IV. METODE KOMPUTASI

A. Model Objek massa jenis

Model objek yang digunakan pada pengujian ini adalah *standard cubic model* (Gambar 3a) [5]. Model berupa satu buah bodi massa jenis dengan massa jenis (ρ) 1 gr/cc dengan dimensi 20 x 20 x 20 meter. Pusat massa dari bodi massa jenis objek tersebut berada pada koordinat (0,0,0) dengan batas bawah $\xi_1 = \eta_1 = \zeta_1 = -10\text{ m}$ dan batas atas $\xi_2 = \eta_2 = \zeta_2 = 10\text{ m}$ seperti yang dapat dilihat pada Gambar 3b. Berdasarkan model bodi massa jenis ini, akan dihitung nilai gravitasi menggunakan kelima metode pengujian pada beberapa titik stasiun yang telah ditentukan.

B. Model Sebaran Posisi Stasiun

Posisi stasiun observasi diletakkan secara grid dalam tiga dimensi yang mewakili segala kemungkinan. Posisi didistribusikan dengan jarak 2 meter satu sama lain terhadap sumbu x, y dan z mulai dari titik paling bawah di $x = -20, y = -20$ dan $z = -20$ hingga titik paling atas di $x = 20, y = 20$ dan $z = 20$ yang dapat dilihat pada Gambar 1b. Total posisi stasiun yang akan dihitung adalah 9261 titik untuk setiap perhitungan.

C. Metode yang Diuji

Penelitian ini mengkaji lima metode *forward modeling* yaitu yang dikembangkan oleh Sorokin, Haaz-Jung-Plouff, Okabe, Nagy dan Zhou. Metode ini akan menghitung nilai gravitasi (g) dengan konstanta gravitasi (G) dan jarak antara stasiun dengan bodi (r) pada posisi pengujian.

1) Metode Sorokin (1951) (Li, Op.Cit. [5]), memiliki persamaan sebagai berikut:

$$g = -G\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times \left[x_i \ln(y_j + r_{ijk}) + y_j \ln(x_i + r_{ijk}) + z_k \arctan \frac{z_k r_{ijk}}{x_i y_j} \right] \quad (5)$$

2) Metode Haaz-Jung-Plouff (1953, 1961, 1966, 1975, 1976) (Li, Op.Cit. [5]), memiliki persamaan sebagai berikut:

$$g = -G\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times \left[x_i \ln(y_j + r_{ijk}) + y_j \ln(x_i + r_{ijk}) + z_k \arctan \frac{x_i y_j}{z_k r_{ijk}} \right] \quad (6)$$

3) Metode Okabe (1979) (Li, Op.Cit. [5]), memiliki persamaan sebagai berikut:

$$g = -G\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times \left[x_i \ln(y_j + r_{ijk}) + y_j \ln(x_i + r_{ijk}) + 2z_k \arctan \frac{x_i + y_j + r_{ijk}}{z_k} \right] \quad (7)$$

4) Metode Nagy (1966) (Li, Op.Cit. [5]), memiliki persamaan sebagai berikut:

$$g = -G\rho \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 \mu_{ijk} \times \left[x_i \ln(y_j + r_{ijk}) + y_j \ln(x_i + r_{ijk}) - z_k \arcsin \frac{y_j^2 + z_k^2 + y_j r_{ijk}}{(y_j + r_{ijk}) \sqrt{y_j^2 + z_k^2}} \right] \quad (8)$$

5) Metode Zhou (2009) [7], memiliki persamaan sebagai berikut:

$$g = G \int_{\zeta_1}^{\zeta_2} \rho(\zeta) \times \left[\tan^{-1} \left(\frac{(\xi - x)(\eta - y)}{(\zeta - z)r_0(\xi, \eta, \zeta)} \right) \right]_{\xi_1}^{\xi_2} \Big|_{\eta_1}^{\eta_2} d\zeta \quad (9)$$

dimana,

$$x_i = x - \xi_i, \quad y_j = y - \eta_j, \quad z_k = z - \zeta_k \quad (10)$$

dan,

$$\mu_{ijk} = (-1)^i (-1)^j (-1)^k \quad (11)$$

serta,

$$f(\xi, \eta) \Big|_{\xi_1}^{\xi_2} \Big|_{\eta_1}^{\eta_2} = f(\xi_2, \eta_2) + f(\xi_1, \eta_1) - f(\xi_2, \eta_1) - f(\xi_1, \eta_2) \quad (12)$$

untuk persamaan (5), (6), (7), (8) dan (9),

D. Arsitektur Komputer

Pengujian dilakukan menggunakan sistem operasi berbasis Unix dengan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam pengujian dapat dilihat pada pada Tabel I dan Tabel II.

Tabel I: Deskripsi perangkat keras yang digunakan

Spesifikasi Perangkat Keras	
Nama	Standard Macbook Pro 13"
Prosesor	2.9 GHz Intel Core i7
Memori	8 GB 1600 MHz DDR3

Tabel II: Deskripsi perangkat lunak yang digunakan

Spesifikasi Perangkat Lunak	
Sistem Operasi	OSX version 10.7.5
Bahasa	C++
Compiler	g++ (MacPorts gcc47 4.7.4_5) 4.7.4

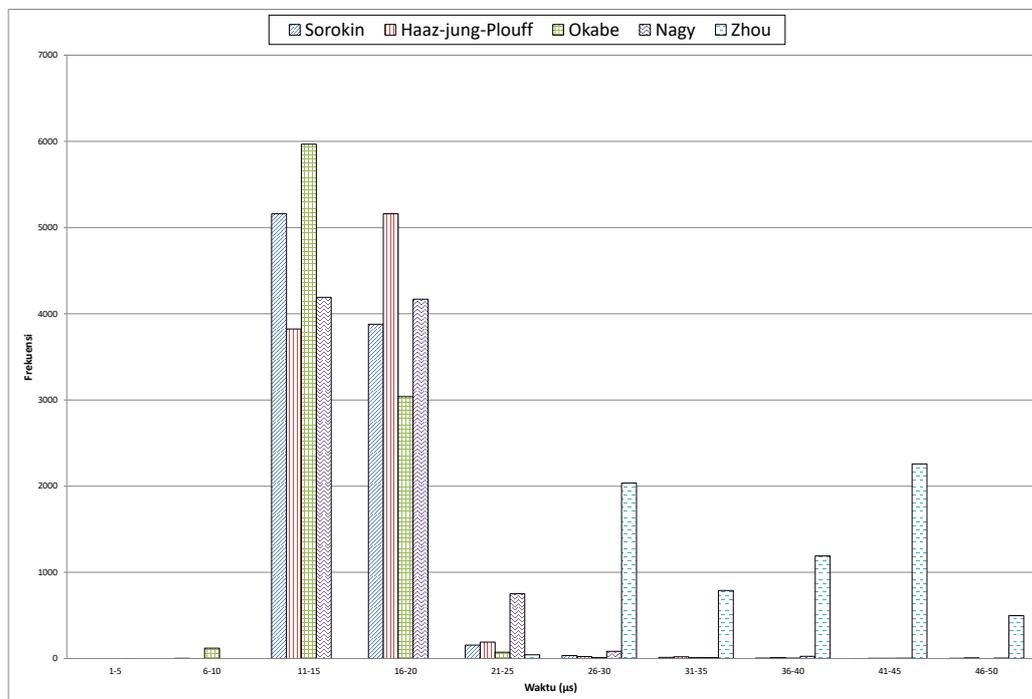
V. HASIL DAN DISKUSI

A. Efisiensi

Waktu perhitungan dari semua metode dapat dilihat pada histogram Gambar 4. Sumbu horizontal menyatakan lamanya waktu untuk menyelesaikan *forward modeling* pada stasiun observasi, sedangkan sumbu vertikal menyatakan frekuensi dari banyaknya titik stasiun observasi yang menyelesaikan perhitungan pada waktu yang ditunjukkan oleh sumbu horizontal. Dari 9261 kali perhitungan, hampir 6000 kali metode Okabe (diwakili oleh diagram warna hijau dengan pola kotak) dapat menyelesaikan perhitungannya selama 11-15 mikrodetik per-stasiun, dan lebih dari 3000 kali selama 16-20 mikrodetik, secara rata-rata metode

ini memiliki kecepatan perhitungan 14.41 mikrodetik per-stasiun. Selanjutnya metode Sorokin (diwakili oleh diagram warna biru dengan pola garis diagonal) memakan waktu sedikit lebih lama yaitu sebanyak 5160 kali selama 11-15 mikrodetik, dan 3879 kali selama 16-20 mikrodetik per-stasiun perhitungan, secara rata-rata metode ini memiliki kecepatan perhitungan 14.97 mikrodetik per-stasiun. Metode Haaz-Jung-Plouff (diwakili oleh diagram warna merah dengan pola garis vertikal) membutuhkan waktu 11-15 mikrodetik sebanyak 3823 kali, 16-20 mikrodetik sebanyak 5161 kali dan 21-25 mikrodetik sebanyak 156 kali, secara rata-rata metode ini memiliki kecepatan perhitungan 16.38 mikrodetik per-stasiun. Kemudian metode Nagy (diwakili oleh diagram warna ungu dengan pola garis bergelombang) memakan waktu 11-15 mikrodetik sebanyak 4191 kali, 16-20 mikrodetik sebanyak 4169 kali dan 21-25 mikrodetik sebanyak 751 kali, secara rata-rata metode ini memiliki kecepatan perhitungan 17.11 mikrodetik per-stasiun. Sedangkan metode Zhou menempati urutan terakhir dengan waktu perhitungan yang paling lama yang tersebar dari 26 hingga 50 mikrodetik lebih untuk tiap perhitungan, secara rata-rata metode ini memiliki kecepatan perhitungan 41.89 mikrodetik per-stasiun.

Hasil ini menempatkan metode Okabe menjadi yang tercepat dari semua metode yang pertama disusun oleh metode Sorokin, metode Nagy, metode Haaz-Jung-Plouff dan baru kemudian metode Zhou menjadi yang terakhir.



Gambar 4: Histogram waktu komputasi untuk menyelesaikan *forward modeling*. Sumbu horizontal menyatakan waktu untuk menyelesaikan *forward modeling* pada stasiun observasi dan sumbu vertikal menyatakan frekuensi dari banyaknya titik stasiun observasi yang menyelesaikan perhitungan pada waktu yang ditunjukkan oleh sumbu horizontal. Warna berpola pada grafik menyatakan metode yang dipakai.

B. Akurasi

Pada percobaan yang dilakukan pada titik kritis seperti yang dapat dilihat pada Tabel III, metode Sorokin memberikan kesalahan yang cukup besar pada beberapa posisi. Jika dibandingkan dengan hasil perhitungan yang dilakukan oleh Li [5], angka yang dihasilkan cukup jauh, hasil ini tidak menunjukkan korelasi yang baik. Sedangkan metode Haaz-Jung-Plouff dan Nagy memberikan hasil perhitungan yang lebih dekat merujuk pada referensi [5], namun sayangnya metode-metode tersebut tidak bisa melakukan perhitungan pada banyak posisi. Ini diindikasikan dengan munculnya *NaN* yang ditampilkan pada hasil perhitungan. Kemudian untuk metode Okabe dan Zhou, tampaknya menghasilkan hasil perhitungan yang

paling baik, dimana hasil perhitungan metode tersebut cukup berkorelasi jika dibandingkan dengan hasil pada referensi. Akan tetapi hasil perhitungan yang diberikan oleh metode Zhou memiliki kesalahan yang sangat signifikan pada posisi stasiun (0,0,0) (namun merujuk pada hasil analisa kapabilitas yang akan dibahas selanjutnya, yang menyatakan bahwa metode Zhou tidak bisa menghitung posisi stasiun di dalam bodi, tampaknya kesalahan perhitungan ini dapat diabaikan). Metode Okabe dan metode Zhou juga menghasilkan hasil perhitungan NaN yang paling sedikit. Hasil NaN pada tabel mengindikasikan dimana terjadinya *singularity*.

Tabel III: Hasil *forward modeling* untuk kelima metode menggunakan bahasa C++ presisi ganda (dc++) dibandingkan dengan referensi yang menggunakan bahasa fortran presisi tunggal (Li-sf) dan ganda (Li-df) [5]. NaN mengindikasikan dimana terjadi *singularity*. Satuan x, y, and z adalah meter, and satuan gaya berat adalah μGal

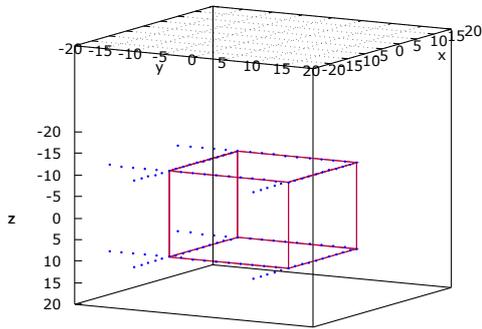
Posisi	x	y	z	Haaz-Jung-Plouff (Li-sf) [5]	Haaz-Jung-Plouff (Li-df) [5]	Sorokin (dc++)	Haaz-Jung Plouff (dc++)	Okabe (dc++)	Nagy (dc++)	Zhou (dc++)
1	0	0	0	0.0000763	0.0000233	0.0000000	0.0000000	0.0000000	0.0000000	838.65947
2	0	0	10	-346.42615	-346.42603	492.0340234	-346.6254452	-346.6254452	-239.1190015	346.625445
3	0	0	100	-5.3354096	-5.3353801	833.3210158	-5.3384528	-5.3384528	-10.5691080	-5.3384528
4	0	0	1000	-0.0533307	-0.05336	838.6060779	-0.0533907	-0.0533907	-0.1067708	-0.0533907
5	0	0	10000	-0.0004071	-0.0005336	838.6589347	-0.0005339	-0.0005339	-0.0010678	-0.0005339
6	0	0	100000	0.0004071	-0.0000053	838.6594633	-0.0000053	-0.0000053	-0.0000107	-0.0000053
7	0	0	1000000	0.0000000	-0.0000001	838.6594686	-0.0000001	-0.0000001	-0.0000001	-0.0000001
8	10	10	10	-129.31641	-129.31636	NaN	NaN	NaN	NaN	NaN
9	100	10	10	-0.518244	-0.5178795	NaN	NaN	-0.5181193	NaN	-0.5181193
10	1000	10	10	0.0032568	-0.0001443	NaN	NaN	-0.0005337	NaN	-0.0005337
11	10000	10	10	0.0521094	0.0000003	NaN	NaN	-0.0000005	NaN	-0.0000005
12	100000	10	10	0.0000000	0.0000000	NaN	NaN	0.0000000	NaN	0.0000000
13	1000000	10	10	0.0000000	0.0000000	NaN	NaN	0.0000000	NaN	0.0000000
14	10	100	10	-0.5178369	-0.5178795	NaN	NaN	-0.5181193	-0.5181193	-0.5181193
15	100	100	10	-0.186861	-0.1872637	-0.1873538	-0.1873538	-0.1873538	-0.1873538	-0.1873538
16	1000	100	10	-0.0065137	-0.0010646	-0.0005259	-0.0005259	-0.0005259	-0.0005259	-0.0005259
17	10000	100	10	0.0000000	0.0064784	0.0000005	-0.0000005	-0.0000005	-0.0000005	-0.0000005
18	100000	100	10	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
19	1000000	100	10	0.0000000	-0.0000042	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

C. Kapabilitas

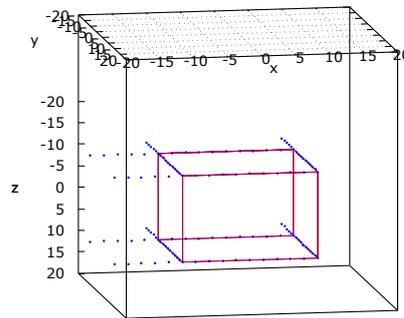
Metode Sorokin, Haaz-Jung-Plouff dan Okabe memiliki kemampuan perhitungan yang sama. Ketiganya mampu menghitung titik stasiun observasi paling banyak. Titik yang mampu dihitung yaitu sejumlah 9141 dari 9261 total stasiun, dan hanya memiliki 120 titik error. Posisi titik stasiun yang tidak mampu dihitung oleh metode ini umumnya berada pada titik yang berhimpit dengan posisi rusuk bodi serta yang sejajar dengannya (Gambar 5a dan 5b, pada gambar tersebut tampak bahwa posisi stasiun yang diwakili oleh titik berwarna biru berhimpit dengan rusuk bodi densitas yang diwakili oleh garis berwarna hitam). Metode Nagy memiliki jumlah stasiun error yang lebih banyak, yaitu sejumlah 952. Posisi yang tidak mampu dihitung oleh metode Nagy umumnya berada pada sisi dari bodi dan yang sejajar dengan bidang tersebut (Gambar 5c dan 5d). Sedangkan metode Zhou memiliki jumlah titik error paling banyak yaitu sejumlah 1129 titik error, terlihat bahwa metode Zhou tidak bisa menghitung pada posisi titik stasiun yang berada di dalam bodi (Gambar 5e dan 5f). Titik stasiun yang tidak bisa dihitung karena adanya *singularity* sebenarnya bisa ditanggulangi dengan menambahkan beberapa kondisi pada persamaan yang ada, namun hal ini tentunya akan menambah waktu komputasi.

VI. KESIMPULAN

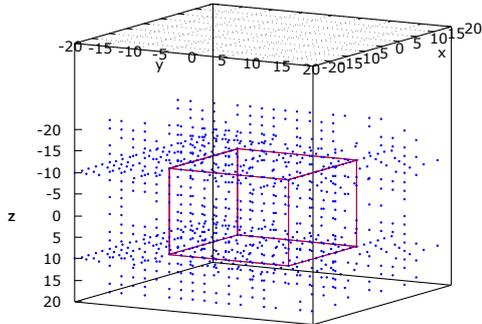
Dalam penelitian ini telah dilakukan pengujian performa lima metode *forward modeling* menggunakan sistem *standard cubic model* dengan 9279 titik stasiun observasi pengujian. Berdasarkan hasil pengujian diperoleh bahwa metode Okabe memiliki waktu perhitungan yang paling cepat dibandingkan empat metode yang lain dengan tingkat efisiensi sebesar 290.70%, kemudian urutan selanjutnya adalah metode Sorokin dengan tingkat efisiensi sebesar 279.77%, disusul oleh metode Haaz-Jung-Plouff dengan tingkat efisiensi sebesar 255.73%, kemudian selanjutnya metode Nagy dengan tingkat efisiensi sebesar 244.84% dan yang terakhir adalah metode Zhou dengan tingkat efisiensi sebesar 100%. Tingkat efisiensi dihitung dengan membandingkan rata-rata waktu perhitungan per-stasiun metode yang memiliki waktu perhitungan



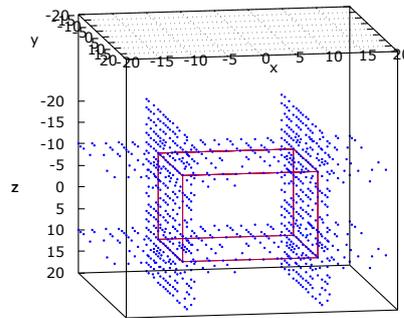
(a) Sorokin, Haaz-Jung-Plouff dan Okabe 1.



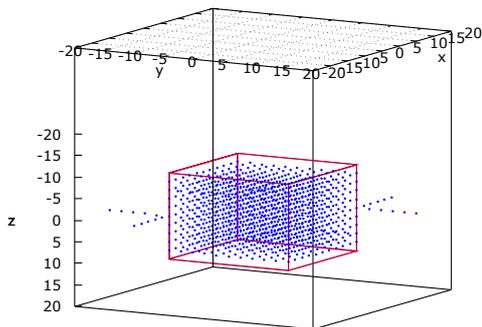
(b) Sorokin, Haaz-Jung-Plouff dan Okabe 2.



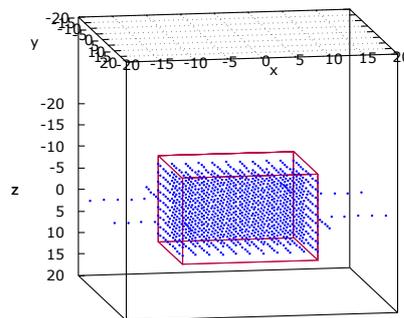
(c) Nagy 1.



(d) Nagy 2.



(e) Zhou 1.



(f) Zhou 1.

Gambar 5: Titik berwarna biru menunjukkan posisi stasiun observasi yang tidak bisa dihitung (terjadinya *singularity*). Perhitungan menggunakan metode Sorokin, Haaz-Jung-Plouff dan Okabe ditunjukkan pada Gambar (a) dan (b), metode Nagy oleh Gambar (c) dan (d) dan metode Zhou oleh Gambar (e) dan (f). Masing-masing dilihat dari dua sudut pandang.

rata-rata paling lama yaitu metode Zhou dengan metode yang sedang diuji. Dari segi akurasi, metode Okabe dan Zhou mendapatkan hasil yang paling baik dengan tingkat korelasi mendekati 1 ketika dibandingkan dengan hasil yang diperoleh oleh Li, dkk. Sedangkan dari segi kapabilitas, metode Sorokin, Haaz-Jung-Plouff dan Okabe memiliki error yang paling sedikit. Rasio error yang dihasilkan adalah 120:120:120:952:1129 dengan urutan perbandingannya adalah metode Sorokin, Haaz-Jung-Plouff, Okabe, Nagy dan Zhou. Berdasarkan hasil tersebut, dapat disimpulkan bahwa dalam kondisi tanpa ada modifikasi persamaan, metode Okabe merupakan metode *forward modeling* yang paling baik dari kelima metode yang diuji.

PUSTAKA

- [1] Wahyudi, E.J., Santoso, D., Kadir, W.G.A., dan Alawiyah, S., "Designing a Genetic Algorithm for Efficient Calculation in Time-Lapse Gravity Inversion", *J.Eng.Technol.Sci.*, 2014, Vol. 46, No. 1, P. 58-77
- [2] Marcotte, D., Shamsipour, P., Olivier, C., dan Chouteau, M., "Inversion of potential fields on nodes for large grids", *Journal of Applied Geophysics*, 2014, 110, P. 90-97
- [3] D'Urso, M.G., "Gravity effects of polyhedral bodies with linearly varying density", *Celest Mech Dyn Astr*, 2014, 120, P.349-372
- [4] Liu, S., Hu, X., Liu, T., Xi, Y., Cai, J., dan Zhang, H., "Ant colony optimisation inversion of surface and borehole magnetic data under lithological constraints", *Journal of Applied Geophysics*, 2015, 112, P. 115-128
- [5] X. Li, dan M. Chouteau, "Three-dimensional gravity modeling in all space." *Surveys in Geophysics*, 1998,19, 339-368.
- [6] D. Plouff, "Gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections." *Geophysics*, 1976, Vol. 41, No. 4, P. 727-741.
- [7] X. Zhou, "3D vector gravity potential and line integrals for the gravity anomaly of a rectangular prism with 3D variable density contrast." *Geophysics*, 2009, Vol. 74, No. 6, P. 143-153.
- [8] M. Commer, "Three-dimensional gravity modelling and focusing inversion using rectangular meshes." *Geophysical Prospecting*, 2011, 59, 966-979.
- [9] "GSL - GNU Scientific Library", Juli 2016, <https://www.gnu.org/software/gsl/>
- [10] "Boost C++ Libraries", Juli 2016, <http://www.boost.org/>